

PISP – Domaća Zadaća

1. Zadatak

Zadatak 1 (30%)

a) Odrediti rezultat koji se ispisuje na ekranu za sljedeće linije kôda:

- 1) `>> y=[10 40 20 30];` `>> x=find(y<30)`
- 2) `>> a=[1 2 3;4 5 6;7 8 9];` `>> diag(fliplr(flipud(a)),-2)`
- 3) `>> n=ones(3,5);` `>> [m]=size(n)`
- 4) `>> A=[1 3 6; 2 5 4];` `>> sum(~(A>3))`
- 5) `>> z=[1-2i; 2+j];` `>> w=z.'`

1-1

```
>> y=[10 40 20 30];
>> x=find(y<30)
```

```
x =
```

```
1    3
```

1-2

```
>> a=[1 2 3;4 5 6;7 8 9];
>> diag(fliplr(flipud(a)),-2)
```

```
ans =
```

```
3
```

1-3

```
>> n=ones(3,5);
>> [m]=size(n)
```

```
m =
```

```
3    5
```

1-4

```
>> A=[1 3 6; 2 5 4];
>> sum(~(A>3))

ans =

     2     1     0
```

1-5

```
>> z=[1-2i; 2+j];
>> w=z.'
```

w =

1.0000 - 2.0000i 2.0000 + 1.0000i

2. Dio prvog zadatka

b) U jednoj liniji programskog kôda napisati rješenja za svaki od sljedećih primjera:

- 1) Koristeći ugrađene funkcije i operator (:) kreirati matricu $A = \begin{bmatrix} 3 & 3 & 3 & 3 \\ 8 & 4 & 2 & 1 \\ 1 & 4 & 16 & 64 \\ 2 & 5 & 8 & 11 \end{bmatrix}$.
- 2) Kreirati skalar S koji određuje broj elemenata druge i treće kolone matrice B koji su manji od elementa koji se nalazi u presjeku dijagonala matrice B dimenzija nxn (n je neparno)
- 3) Koristeći `nonzeros` funkciju i linearno indeksiranje matrica kreirati vektor X koji sadrži parne prirodne elemente na neparnim indeksima matrice C.

```
A=[linspace(3,3,4); 2.^(3:-1:0); 4.^(0:3); 2:3:11];
S= length(find(B(:,2:3)<B(ceil(n/2),ceil(n/2))));
X = nonzeros(C(1:2:end) .* (mod(C(1:2:end), 2) == 0));
```

2. Zadatak

Zadatak 2 (35%)

Napisati skriptu `z2.m` koja ponavlja naredbu za unos matrice sve dok korisnik ne unese matricu A dimenzija $n \times n$ ($n \geq 3$).

U slučaju da su elementi matrice A raspoređeni tako da matrica predstavlja magični kvadrat (ukupni zbir elemenata po svakom redu, koloni i dijagonalama je isti) skripta ispisuje odgovor DA i vektor V čiji su redom elementi:

- proizvod elemenata sa dijagonale ispod sporedne dijagonale matrice A i elemenata sa dijagonale iznad glavne dijagonale matrice A
- suma parnih elemenata sa glavne i sporedne dijagonale submatrice matrice A (u slučaju da se paran broj nalazi na presjeku dijagonala treba ga samo jednom računati)
- broj rubnih elemenata matrice A koji su manji od 8, a čiji je zbir indeksa na kojem se nalazi element neparan broj.

U suprotnom slučaju skripta ispisuje odgovor NE i matricu B koja sadrži iste elemente po dijagonalama kao matrica A, a ostali elementi su nasumično kreirani prirodni brojevi iz intervala $[2,8]$.

```
while true A = input('Unesite matricu A: '); [n, m] = size(A);

if n ~= m || n < 3
    disp('Matrica mora biti kvadratna i dimenzija >= 3. ');
    continue;
end

n = length(A);
A_lr = fliplr(A);

sume = [sum(A,1) sum(A,2)' sum(diag(A)) sum(diag(A_lr))];
if ~length(find(sume ~= sume(1)))
    disp('DA');

    prod_val = prod(diag(A,1)) * prod(diag(flipud(A), -1));
    D = prod_val;

    d1 = diag(A);
    d2 = diag(flipud(A));
    if mod(n,2) == 1
        d2(ceil(n/2)) = [];
    end
    suma_parnih = sum([d1; d2] .* (mod([d1; d2],2) == 0));
```

```

B = suma_parnih

broj_rubnih=0;
for i=1:n
    for j=[1 n]
        if A(i,j)<8 && mod(i+j,2) ~= 0
            broj_rubnih =broj_rubnih+1;
        end
    end
end

for i=[1 n]
    for j=2:(n-1)
        if A(i,j)<8 && mod(i+j,2) ~= 0
            broj_rubnih=broj_rubnih+1;
        end
    end
end

C= [];
C = broj_rubnih;

% Finalni vektor
V = [D B C];
disp('Vektor V:');
disp(V);
else
    disp('NE');
    B=round(rand(n)*6+2);
    a2 = diag(diag(A));
    b2 = diag(diag(B));
    a3 = flipud(diag(diag(flipud(A))));
    b3 = flipud(diag(diag(flipud(B))));
    B = B - b2 - b3 + a2 + a3;
    disp('Matrica B: ');
    disp(B);
end
end

```

3. Zadatak

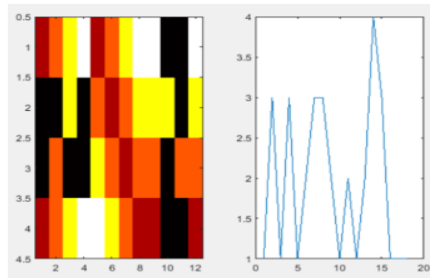
Zadatak 3 (35%)

Napisati MATLAB funkciju `z3.m` kod koje je ulaz vektor-red `x` sa minimalno 3 elementa, a izlaz matrica `A`. Funkcija kreira matricu `A` prema sljedećem uzorku:

```
>> x=[1 2 3 4];
>> A=z3(x)
A =
     1     2     3     4     1     2     3     4     4     0     0     4
     0     0     3     0     2     1     2     3     3     3     0     3
     0     2     0     0     3     2     1     2     2     0     2     2
     1     2     3     4     4     3     2     1     1     0     0     1
```

Nakon kreiranja matrice `A`, funkcija ispisuje na ekranu svaki element matrice sa sumom njegovih najbližih susjednih elemenata koji ga okružuju (koristeći funkciju `conv2`) npr.

```
A(1,1)=1 okružen sumom 2
A(1,2)=2 okružen sumom 7
A(1,3)=3 okružen sumom 9
A(1,4)=4 okružen sumom 9
A(1,5)=1 okružen sumom 9
A(1,6)=2 okružen sumom 9
A(1,7)=3 okružen sumom 12
A(1,8)=4 okružen sumom 15
A(1,9)=4 okružen sumom 13
A(1,10)=0 okružen sumom 10 itd.
```



Funkcija crta u jednom grafičkom prozoru u lijevom axis-u sliku matrice `A` sa crno-crveno-žuto-bijelom paletom boja, a u desnom axis-u vektor "djelilaca" matrice `A` (element matrice različit od 0 koji je djelilac sume njegovih najbližih susjednih elemenata koji ga okružuju). U ostalim slučajevima funkcija javlja grešku.

```
function A = z3(x)
```

```
n = length(x);
```

```
A = zeros(n, 3*n);
```

```
% Prvi blok – sporedna dijagonala sa x
```

```
A(end:-1:1, 1:n) = diag(x);
```

```
% Treci blok - dijagonala
```

```
A(1:n, 2n+1:3n)=diag(fliplr(x));
```

```
% Drugi blok – dijagonala sa 1 i iznad nje sa 2 i ispod nje sa 2
```

```
A(:, n+1:2n) = diag(ones(1, n)) + diag(2*ones(1, n-1), 1) + diag(2*ones(1, n-1), -1);
```

```
% Treci blok – prva i zadnja kolona x i obrnuti x
```

```
A(1:n, 2n+1) = x(end:-1:1);
```

```
% prva kolona treceg bloka
```

```
A(1:n, 3n) = x(end:-1:1);
```

```
% zadnja kolona treceg bloka
```

```

% Rubni redovi i kolone – gornji, donji, lijevi, desni
A(1, 1:n) = x;
% gornji red prvog bloka
A(n, 1:n) = x;
% donji red prvog bloka
A(1:n, n+1) = x;
% prva kolona srednjeg bloka
A(1, n+1:2n) = x;
% gornji red srednjeg bloka
A(n, n+1:2n) = x(end:-1:1);
% donji red srednjeg bloka
A(1:n, 2*n) = x(end:-1:1);
% zadnja kolona srednjeg bloka

maska = ones(3);
maska(2,2) = 0;
% izuzmi centar
sume = conv2(A, maska, 'same');

% Ispis svakog elementa i njegove okoline
[rows, cols] = size(A);
for i = 1:rows
    for j = 1:cols
        fprintf('A(%d,%d)=%d okruzen sumom %d\n', i, j, A(i,j),
            sume(i,j));
    end
end

% Djelilac suma susjeda
djelilac_maska = (A ~= 0) & mod(sume, A) == 0;
D = A(djelilac_maska);

% Crtanje slika
figure;

% Lijeva slika: slika matrice A
subplot(1,2,1);
imagesc(A);
colormap(gca, hot);

```

```
colorbar;  
title('Matrica A');  
  
% Desna slika: vektor djelilaca  
subplot(1,2,2);  
plot(D, '-o');  
title('Djelilac suma susjeda');  
xlabel('Indeks');  
ylabel('Vrijednost');  
  
end
```